

# 5DT Data Glove 16

## 5DT Data Glove 16-W



*Data gloves for the fifth dimension*

**User's Manual**

**5DT**

<Fifth Dimension Technologies>

# 5DT Data Glove 16

16-sensor\* data glove

# 5DT Data Glove 16-W

Wireless 16-sensor\* data glove

*Data gloves for the fifth dimension*

# User's Manual

All rights reserved. No part of this manual, software or hardware may be copied, reproduced, translated or reduced to any electronic medium or machine readable format without the written consent of 5DT <Fifth Dimension Technologies>. The owner of this product may however make copies of the software for backup purposes.

Windows is a registered trademark of Microsoft Corporation. All other trademarks and tradenames are the respective property of their owners.

Unless otherwise indicated, the term "5DT Data Glove 16" in this manual refers to both the standard and the wireless data glove. The wireless glove will be indicated as "5DT Data Glove 16-W" in circumstances where there are differences between the two devices.

*\* 14 Sensors implemented as on 31 January 2000*

# Table of Contents

---

1 INTRODUCTION .....	1
2 SETUP AND INSTALLATION.....	2
2.1 PACKAGE CONTENTS .....	2
2.2 CONNECTING THE STANDARD GLOVE.....	4
2.3 CONNECTING THE WIRELESS GLOVE.....	6
2.4 SOFTWARE INSTALLATION .....	8
3 GLOVE MANAGER .....	10
3.1 RUNNING GLOVE MANAGER .....	10
3.2 CONNECTING THE GLOVE TO THE COMPUTER.....	12
3.3 THE SOFTWARE CALIBRATION ROUTINE .....	13
3.4 SAVING THE GLOVE DATA TO A LOG FILE .....	14
3.5 CHANGING THE PROGRAM OPTIONS .....	15
4 OTHER SOFTWARE.....	16
4.1 GLOVE DRIVER .....	16
4.2 GLOVE DEMOS.....	16
4.3 GLOVE PROGRAMMING EXAMPLES.....	18
5 TROUBLESHOOTING AND SUPPORT .....	19
5.1 GENERAL TROUBLESHOOTING.....	19
5.2 FREQUENTLY ASKED QUESTIONS (FAQ).....	20
5.3 SUPPORT.....	21
6 WARRANTY INFORMATION .....	22
6.1 THIRTY (30) DAY CUSTOMER SATISFACTION GUARANTEE .....	22
6.2 ONE (1) YEAR PRODUCT WARRANTY.....	22
6.3 WARRANTY CLAIM INSTRUCTIONS .....	22
7 ABOUT 5DT.....	23
APPENDIX A - HARDWARE SPECIFICATIONS .....	25
APPENDIX B - INTERFACE CABLE TECHNICAL DETAILS.....	26
APPENDIX C - SERIAL PORT INTERFACE PROTOCOL.....	27
C1. COMMUNICATIONS.....	27
APPENDIX D – 5DT DATA GLOVE DRIVER.....	29
1. INTRODUCTION.....	29
2. USING THE GLOVE DRIVER.....	29
3. SUPPORTED GLOVES .....	31
4. FUNCTION REFERENCE.....	33
5. GESTURE DEFINITIONS.....	44
6. AUTO-CALIBRATION .....	47
APPENDIX E – SOFTWARE CALIBRATION .....	48

# 1 Introduction

---

Thank you for choosing this product. We trust that you will have a lot of fun and satisfaction using it.

The 5DT Data Glove 16 measures finger flexure as well as the abduction between the fingers. Finger flexure is measured at two places (1<sup>st</sup> joint (knuckle), 2<sup>nd</sup> joint) on each finger.

Although every effort has been made to ensure that this is a rugged product, it must still be treated as a sensitive instrument. Do not leave it in the same room as your pet velociraptor or chihuahua.

## Features

- Affordable quality
- One size fits many
- Extreme comfort
- Automatic calibration, minimum 8-bit flexure resolution
- Platform independent - serial port interface (RS 232)
- Bundled software
- High update rate
- Wireless version available (5DT Data Glove 16-W)
- Low crosstalk between fingers
- On-board processor
- Quick "hot release" connection
- Cross-platform drivers

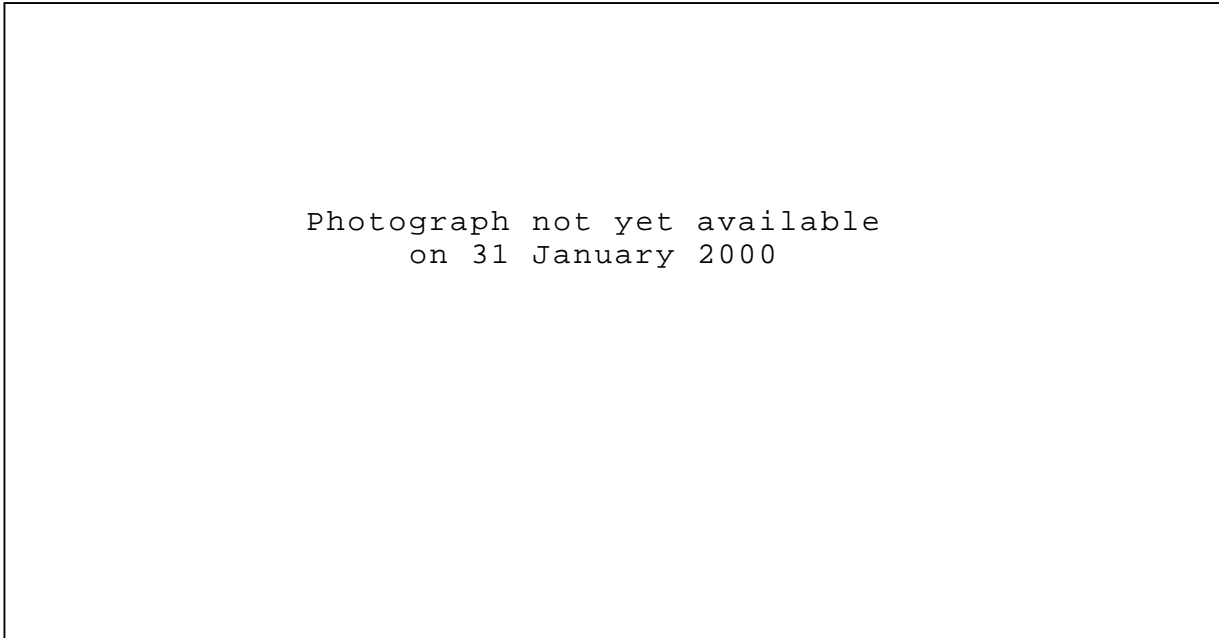
## 2 Setup and Installation

---

Please check to see that your 5DT Data Glove package contains all the necessary items. It will be difficult to properly install and use the product if any item is missing. To connect the glove to your computer, follow the instructions in the relevant section.

### 2.1 Package contents

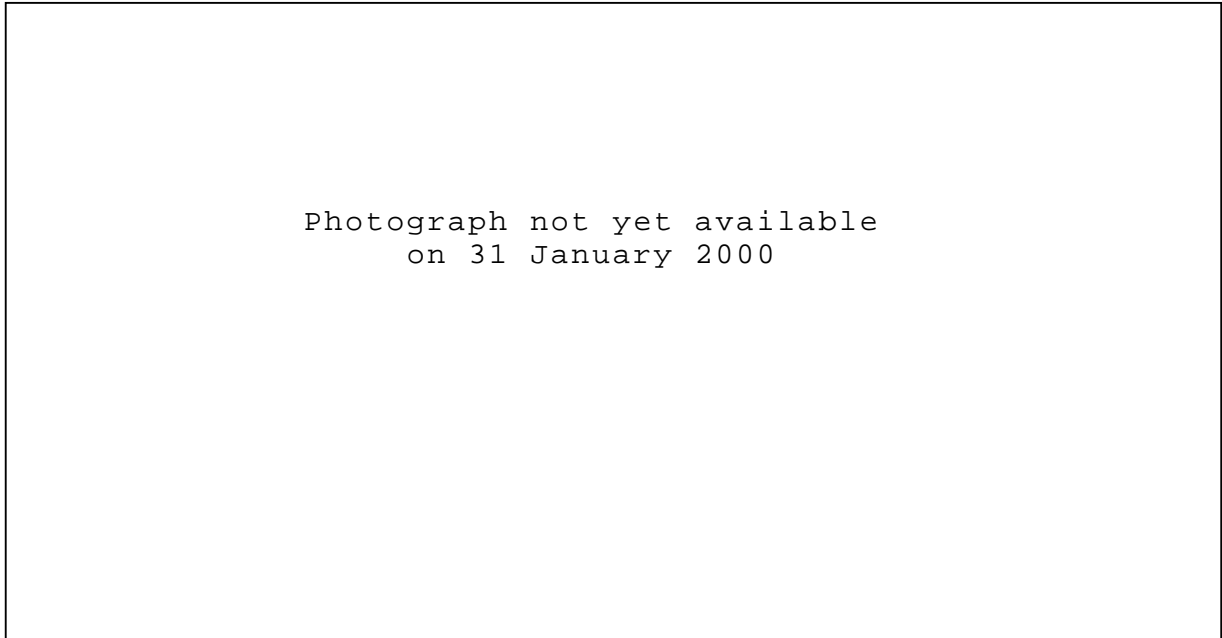
The standard **5DT Data Glove 16** package consists of the following items:



**Figure 1 – Package contents for the 5DT Data Glove 16**

1. The 5DT Data Glove 16 with fiber optic sensors, ribbon cable and processing unit
2. The interface cable
3. A power supply unit
4. A set of installation disks and/or CD-ROM
5. This manual

The package for the wireless **5DT Data Glove 16-W** consists of the following items:



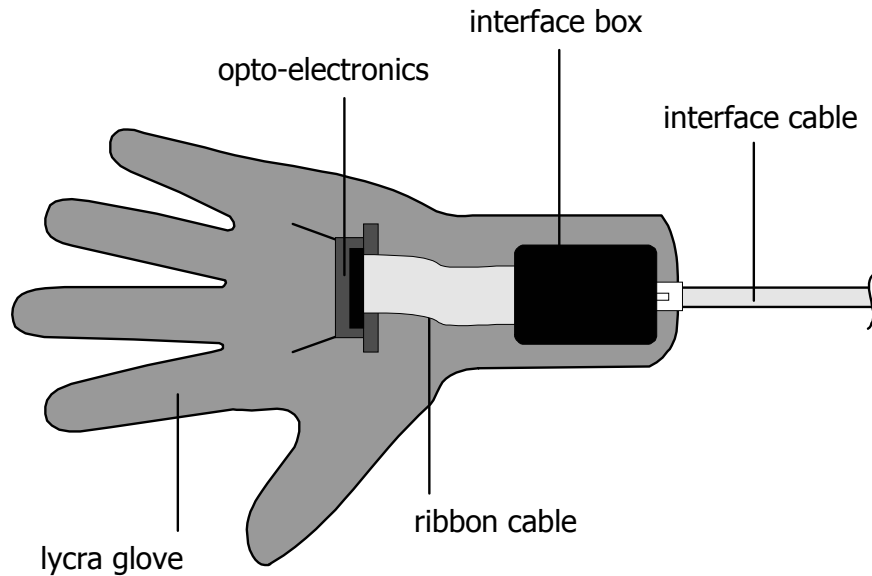
**Figure 2 - Package contents for the 5DT Data Glove 16-W**

1. The 5DT Data Glove 16-W with fiber optic sensors, tilt sensor, ribbon cable and processing unit
2. The interface cable
3. A power supply unit
4. Wireless receiver unit
5. Wireless transmitter unit
6. Short transmitter interface cable
7. Long transmitter interface cable
8. This manual
9. A set of installation disks and/or CD-ROM

Please note that batteries are **not** included in the package.

## 2.2 Connecting the standard glove

The 5DT Data Glove 16 consists of a **lycra glove** with embedded fiber optic sensors. These sensors are linked to the computer via an **opto-electronics** unit, a **ribbon cable** and an **interface box**.



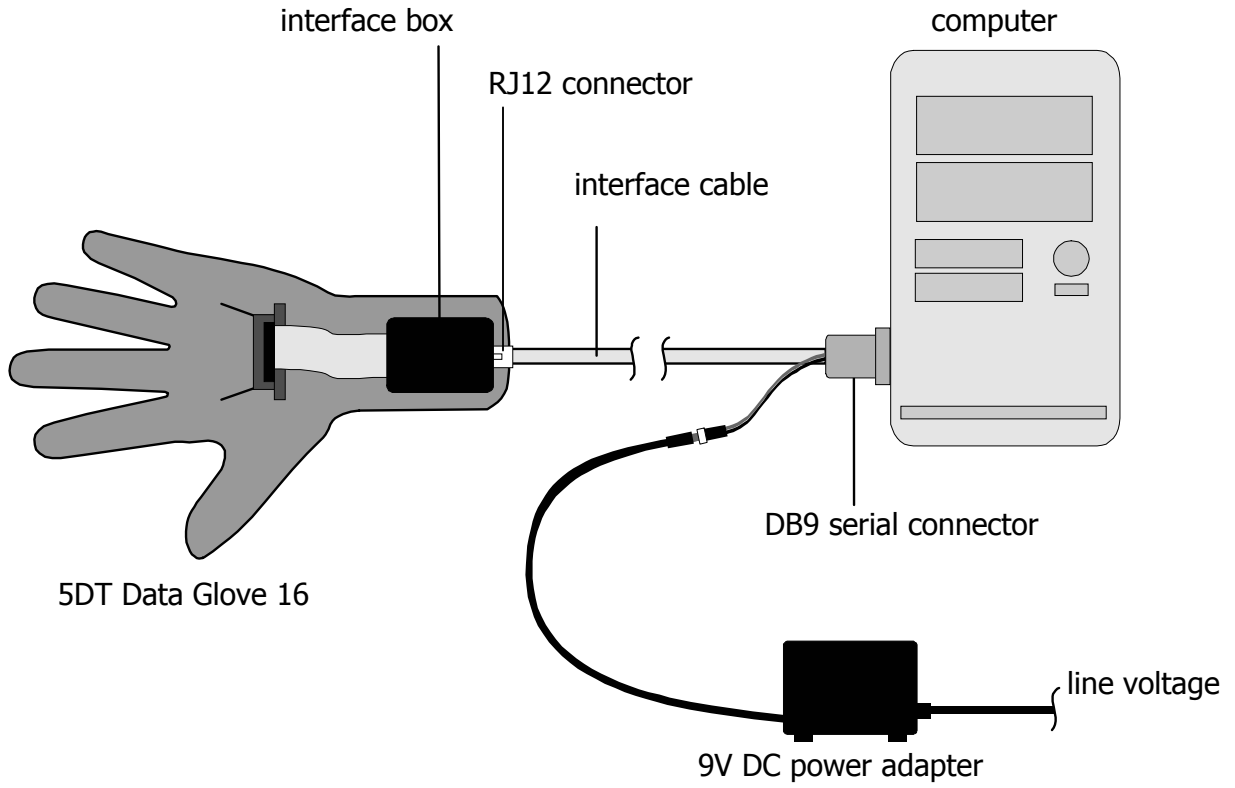
**Figure 3 - The components of the 5DT Data Glove 16**

The 5DT Data Glove 16 connects to a 9-pin RS 232 serial port (DB9 connector) via the interface cable. If the glove is to be used with a 25-pin serial port connector, a 9 to 25-pin adapter may be used.

The recommended procedure for connecting the standard glove to the computer is as follows (please refer to Fig. 4):

1. Connect the serial cable to the computer (DB9 connector).
2. Connect the power supply to the serial interface / power cable.
3. Connect the serial cable to the glove (RJ12 connector).

The following diagram shows the glove connected to the computer:



**Figure 4 - The standard glove connected to the computer**



## 2.3 Connecting the wireless glove

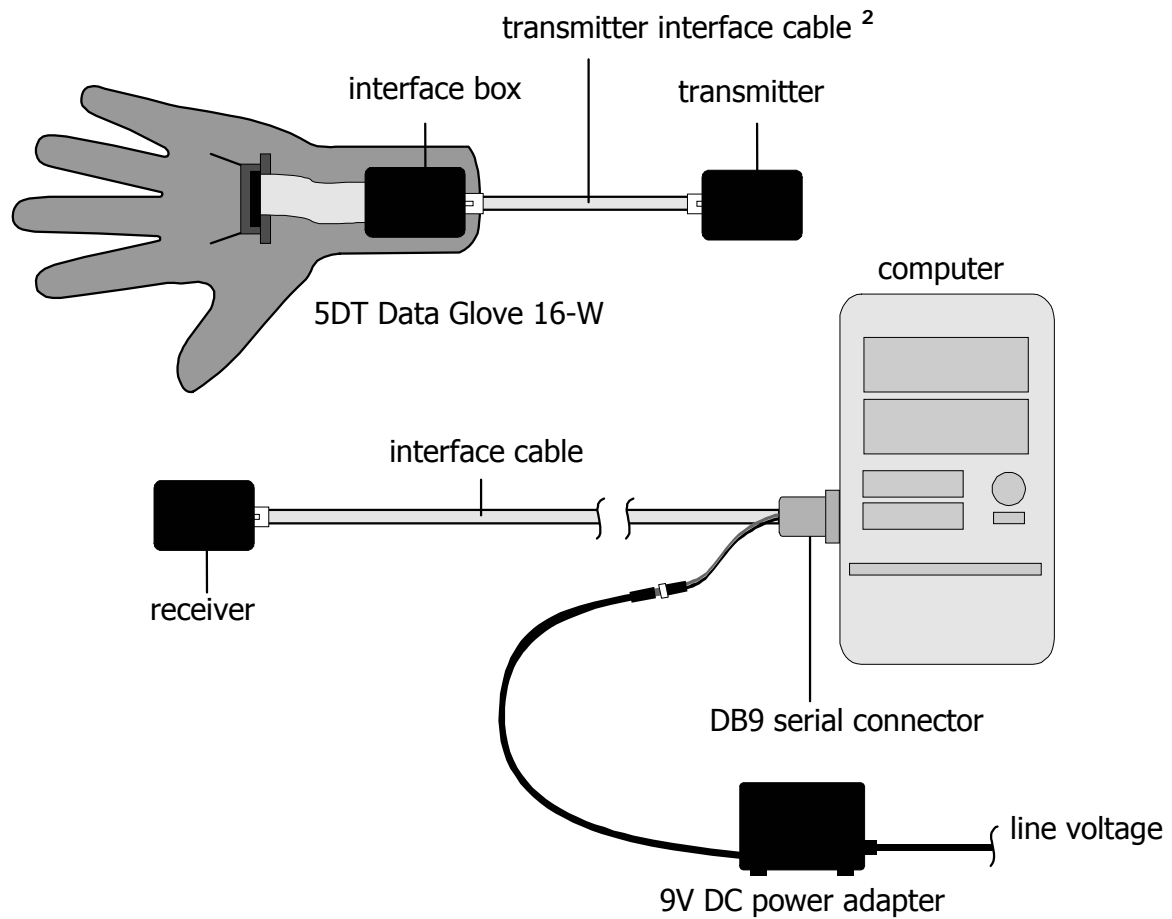
The wireless 5DT Data Glove 16-W is connected to the computer via a transmitter and a receiver unit. The recommended procedure for connecting a wireless glove to the computer is as follows (please refer to Fig. 5):

1. Connect the serial interface cable to the computer (DB9 connector).
2. Connect the power supply to the serial interface / power cable.
3. Connect the serial cable to the receiver unit (RJ12 connector).
4. Place a battery in the transmitter unit (PP3 9V).
5. Connect either the short or the long transmitter cable between the glove interface box and the transmitter unit.
6. Switch the transmitter unit on.<sup>1</sup>

---

<sup>1</sup> Always remember to switch the transmitter off after use!

The following diagram shows the wireless glove connected to the computer:



**Figure 5 - The wireless glove connected to the computer**

---

<sup>2</sup> The transmitter interface cable comes in two lengths; a short and a longer cable. The short cable is intended for use when the transmitter unit is strapped to the wrist. The longer cable enables the user to put the transmitter unit in his/her pocket.

## 2.4 Software installation

The 5DT Data Glove 16 is supplied with a CD-ROM and/or a set of 3½" disks. These disks contain calibration and testing software, applications, drivers, demos, programming examples and this manual in electronic form.

Most of the supplied programs and software are intended for 32-bit Windows, and require Windows 95/98 or Windows NT.

To install the software onto your computer, insert the first disk (or the CD) into your disk drive, run the program `setup.exe` and follow the on-screen instructions.

The suggested installation folder is,

```
c:\Program Files\5DT\Data Glove 16\
```

The software may however be installed in any folder of your choice.

You will be prompted to insert the other disks as they are required. After installation, you may use the setup program to create shortcuts, in your Start menu and/or on your desktop, to the programs and documents that were installed.

Please take a careful look at the `Readme.txt` file in the main installation folder. This file explains the folders that were created during installation, and lists the latest information.

The software installed on your hard drive during installation includes:

- Glove Manager (more detail in section 3)  
...\`GloveMan`
- Glove Driver (more detail in Appendix D)  
...\`Driver`
- Glove Demo (more detail in section 4.2)  
...\`Demo`
- Glove Programming Examples (more details in section 4.3)  
...\`Programming Examples\VC Sample`
- Glove Manual  
...\`Manual`

The default installation folder for shortcuts is:

```
...\Windows\Start Menu\Programs\Data Glove 16
```

The shortcuts that will be created in this folder are:

- Glove Manager
- Demo
- Manual

The CD-ROM also contains the redistributable version of DirectX 7.0. You will need this if you want to run the DirectX implementations of the Glove Demos (DirectX and OpenGL demos are supplied). To install DirectX 7.0, open the

`...\DirectX7.0\`

folder on your CD-ROM drive and run `DXSetup.exe`.

*If you do not have the CD-ROM, you will need to download DirectX 7.0 from the Microsoft website ([www.microsoft.com/directx/](http://www.microsoft.com/directx/)).*

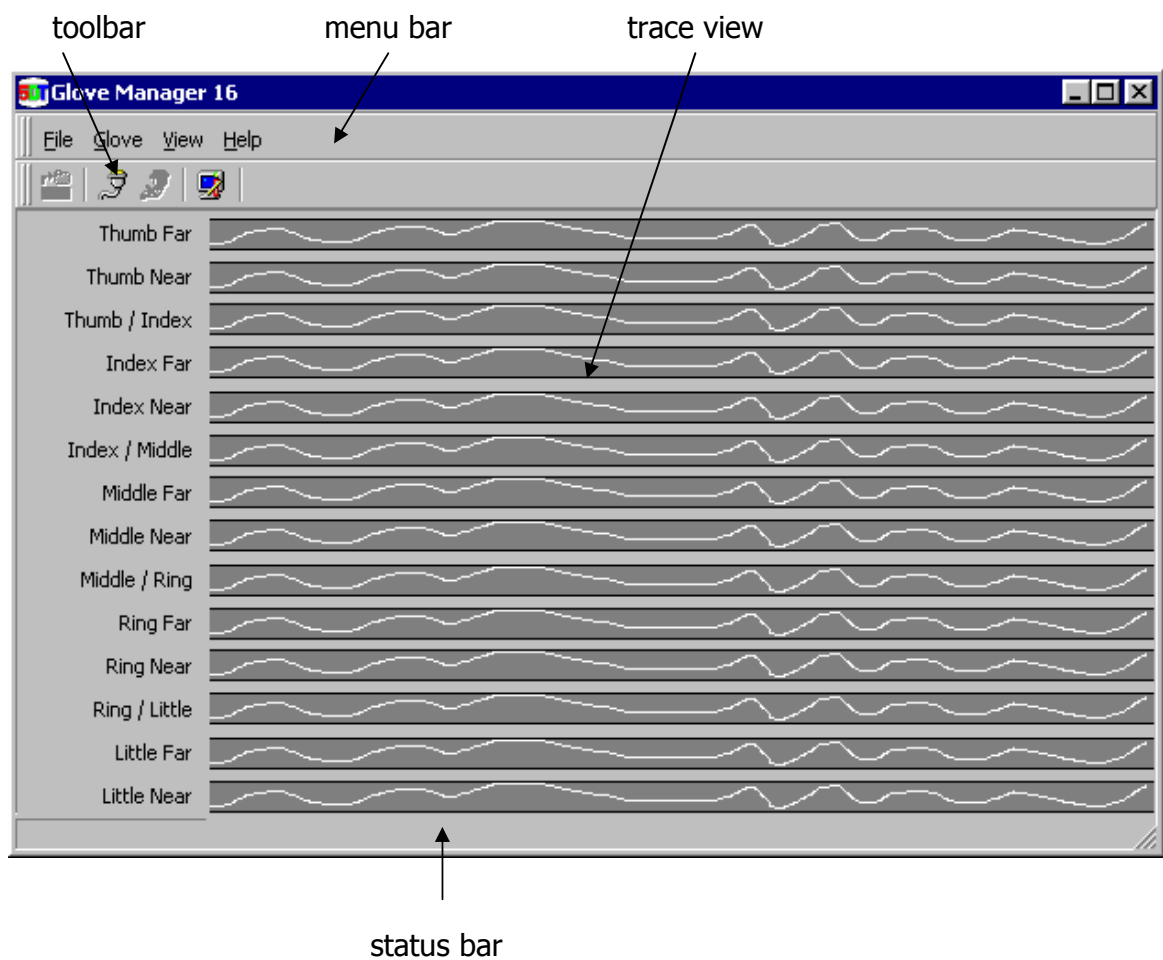
### 3 Glove Manager

The 5DT Glove Manager is a useful program that may be utilized for the following:

- *Connecting* the 5DT Data Glove to the computer
- *Testing* the 5DT Data Glove
- Illustrating a *software calibration* routine for the data glove
- Illustrating a *data logging* routine for the data glove

#### 3.1 Running Glove Manager











Start the program by running `GloveMan.exe`, which is installed in the `...\Data Glove 16\GloveMan\` folder by default. The main program window will appear:



**Figure 6 - The Glove Manager main window**

The top of the window hosts a **menu** and a **toolbar**, and there is a **status bar** at the bottom. The **trace view** shows graphs of the flexure of the fourteen implemented sensors.

The menu and toolbar provide access to all the program's functions. The following table shows the menu structure, and indicates what each icon is used for:

Icon	Menu Command	Description	Shortcut
	File Open Calibration	Read the calibration information from an existing calibration file.	F3
	File Save Calibration	Save the current calibration information to a file.	F2
	File Save Data	Save the data received from the glove to a log file.	F4
	File Exit	Close the program.	Alt+F4
	Glove Connect	Establish communication with the glove via a COM port.	F5
	Glove Info	Show the information returned by the glove, e.g. Version	Ctrl+I
	Glove Calibrate	Calibrate the glove for a certain user's hand.	F8
	Glove Disconnect	Disconnect the glove.	F12
	Raw Data	Shows raw data as it is received from the glove.	Ctrl+R
	Scaled Data	Shows data that has been scaled with the calibration routine.	Ctrl+S
	View Options	View and set the program settings and display options.	F9
	Help About	Display program information.	

**Table 1 - Glove Manager menu structure and icons**

### 3.2 Connecting the glove to the computer

Before you can use the Data Glove, you need to connect to it. Choose **Connect** from the **Glove** menu, or click on the corresponding toolbar icon. The connect dialog box will appear, showing a list of available ports. Select the port (e.g. COM2) where the glove is connected and then click the **Connect** button.

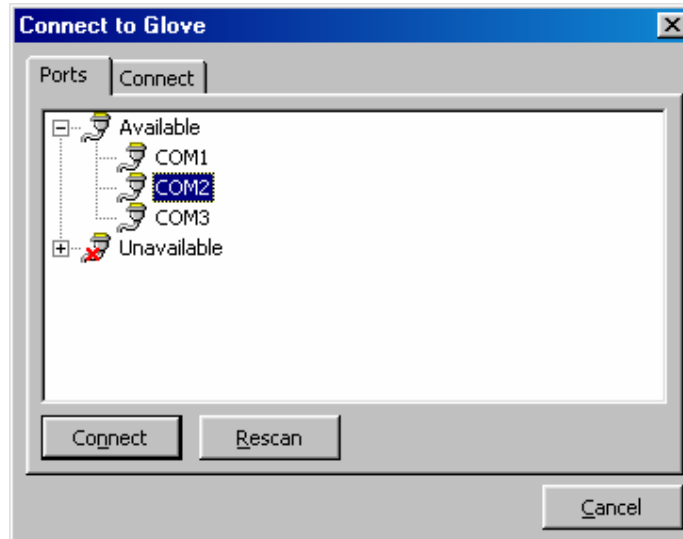


Figure 7 - Glove Manager's connect dialog box

The program will switch from the **Ports** tab to the **Connect** tab and then attempt to open the port and connect to the glove. The following message will be shown (or a slight variation):

```
Opening port COM2...
COM2 opened. Detecting Data Glove 16...
Data Glove 16 found.
Connected to Glove.
```

If a glove cannot be found, the exact reason for this will be displayed. See the troubleshooting section in Chapter 5 for help.

Click the **Close** button to shut the connect dialog box. In the main window, the graphs will show the current flexure of the sensors. As you flex and unflex your fingers, the traces on the flexure graphs will move higher and lower respectively.

### 3.3 The software calibration routine

Each person has a different size hand. The 5DT Data Glove is designed to fit most users comfortably, but it needs to be calibrated in order to achieve the maximum sensitivity within an application.

*It is important to note that this software calibration routine **only** calibrates the glove for use within the Glove Manager program. It serves as an illustration/example only and it does not calibrate the glove for other programs/applications. You will have to develop a software calibration routine of your own for your applications. The process of doing so is described in Appendix E*

*Please note that the supplied Glove Driver features a built-in software calibration routine (Appendix D).*

To calibrate the glove for use within Glove Manager, select **Calibrate** from the **Glove** menu, or click the corresponding toolbar icon. The calibration dialog box will appear:

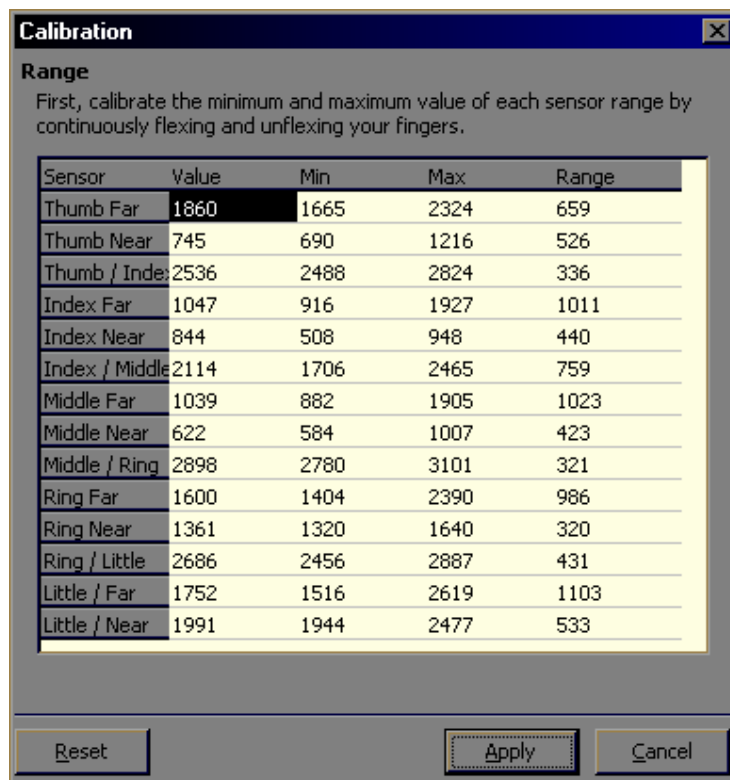


Figure 8 - Calibrating the glove: Range Setup

Repeatedly flex and unflex each of your fingers. It is important to keep your hand **relaxed** at all times, keeping the motion natural. **Do not force your hand**. As you move your fingers, the minimum and maximum values of each of the sensors will be shown in the respective column, and the dynamic range in the last column. The current value of each sensor is shown in the first column.

Once you are satisfied that the minimum and maximum values have been set, click the **Apply** button to end.

To save you time in the future, you may want to save the calibration information to a file. You may then load the calibration at any time very quickly. Select **Save Calibration** or **Load Calibration** from the **File** menu of the main window.



### 3.4 Saving the glove data to a log file

There are situations where you may want to "record" the data received from the glove by saving it to a log file.

Glove Manager 16 can save the glove data in CSV (Comma Separated Values) format. The CSV structure is very simple, and files in this format can be read into programs such as Microsoft Excel very easily.

To save the glove data to a file, open Glove Manager and choose **Save Data** from the **File** menu. The following dialog box will appear:

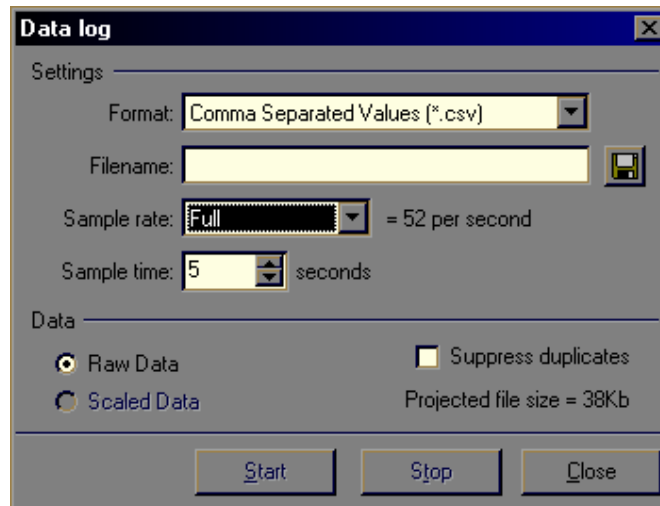


Figure 9 - Glove Manager's data log dialog box

It is possible to change the following settings:

- |                    |   |
|--------------------|---|
| <b>Format</b>      | Only CSV format is available at this time.  |
| <b>Filename</b>    | Enter the name of the file you want to create. You can also click the disk icon to the right of the filename edit box. This will open a save dialog box.  |
| <b>Sample rate</b> | Here you can select Full, Half or Quarter. Selecting Full will save each data packet received from the glove to the log file. The Half setting will only save every second packet, and the Quarter setting will save every fourth packet. |
| <b>Sample time</b> | Enter the maximum number of seconds you would like to record. You can always click the Stop button to stop recording before this maximum time has expired.  |
| <b>Raw data</b>    | Select this option if you would like to save the data exactly as it is received from the glove.   |
| <b>Scaled data</b> | Select this option if you would like to save scaled (calibrated) data to the log file. This option is only available if the user has run the calibration procedure, or loaded a calibration file.   |

To start saving the data to the log file, click the **Start** button. Recording will stop automatically after the specified time interval, or when the **Stop** button has been clicked.

The last option is the **Suppress Duplicates** checkbox. If you select this option, then the program will only output a data value if it has changed. You will want to select this option if you have developed your own processing software and wish to take advantage of the saving in disk space, and quicker data processing times. The following two CSV output samples show the difference between the two settings (*please note that all of the 14 sensors' values will be recorded, Listing 1 serves as an illustration only*):

Duplicate Suppression "off"	Duplicate Suppression "on"
5DT Data Glove Log File DataType=Raw Timestamp,S1,S2,S3,S4,S5,...	5DT Data Glove Log File DataType=Raw Timestamp,S1,S2,S3,S4,S5
0,136,93,96,83,99,...	0,136,93,96,83,99,...
3,134,93,94,83,98,...	3,134,,94,,98,...
7,134,92,93,82,97,....	7,,92,93,82,97,....
10,131,91,93,81,96,...	10,131,91,,81,96,...
14,129,90,92,81,95,...	14,129,90,92,,95,...
17,126,89,92,80,95,...	17,126,89,,80,....
21,124,88,91,80,92,...	21,124,88,91,,92,....
24,120,88,91,80,92,...	24,120,,,,, ...
28,120,88,91,81,92,...	28,,,,81, ...
31,118,88,92,82,92,...	31,118,,92,82, ...
35,117,89,94,82,93,...	35,117,89,94,,93,...
39,117,91,97,83,94,...	39,,91,97,83,94,...

**Listing 1 - CSV File with suppression off and on**

The program does not output a value if it is the same as the previous sample *of the corresponding finger*. Some programs will not be able to interpret the data if the duplicates have been suppressed.

### 3.5 Changing the program options

The Glove Manager has many options you can set to change how it works. To change them start Glove Manager, select **View**, then **Options**. The options dialog box will appear. These are some of the options you can adjust:

#### **Close connect dialog box on successful connection**

Check this option to force the connect dialog box to be automatically closed when a successful connection is made with a glove.

#### **Show toolbar hints**

Check this option to enable the display of popup toolbar hints. These hints explain what each icon represents, and list the relevant shortcut key for that command.

You can change the appearance of the program by selecting one of the predefined **color schemes**. It is also possible to create your own scheme by adjusting the colors.

## 4 Other Software

---

Please consult the `Readme.txt` file in the main installation folder. This file fully explains all the software that was installed on your hard drive.

### 4.1 Glove Driver

`...\Driver`

The 5DT Data Glove Driver is explained in detail in Appendix D.

### 4.2 Glove Demos

`...\Demo`

These Demos currently allow you to do the following:

- View the animated hand on its own
- Use the animated hand to navigate a landscape
- Animate the movement of a character
- Animate the movement of a character in a landscape

The Demo screen is shown in Fig. 10

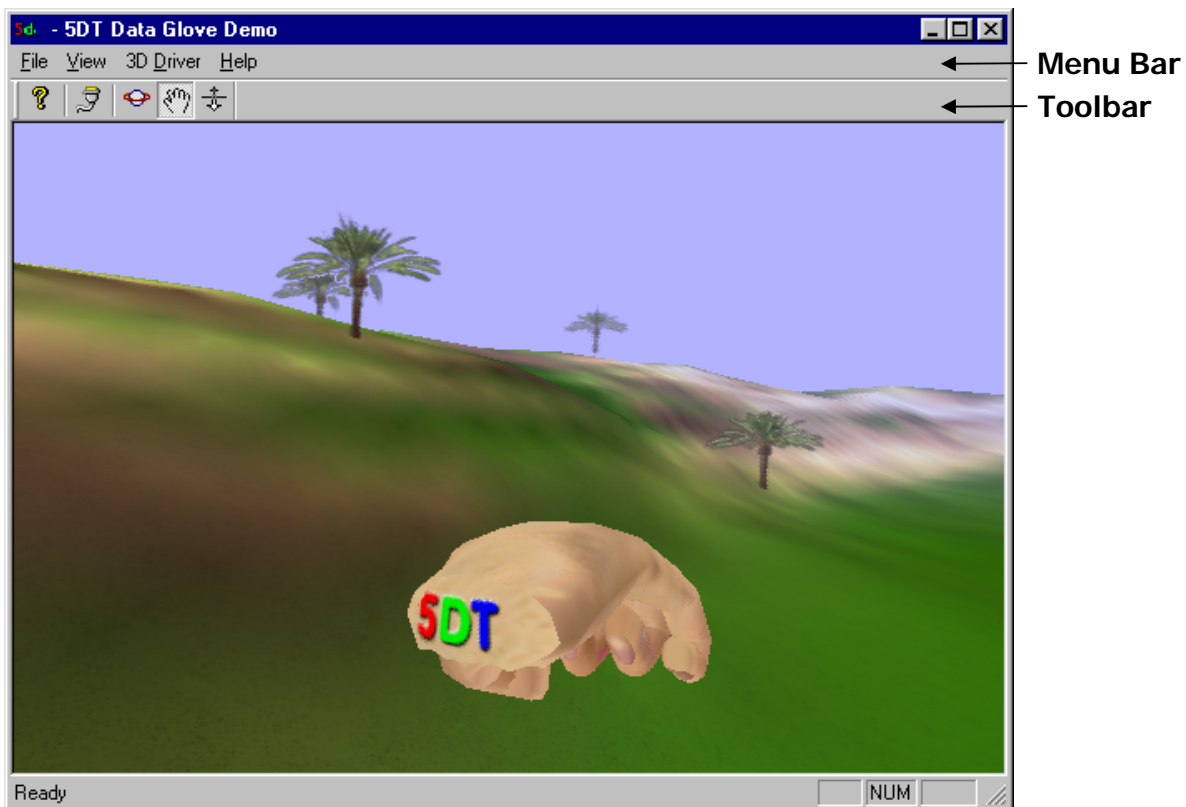


Figure 10 – Screen Shot of 5DT Data Glove 16 Demo

The following menu bar options are available:

## **File**

Exit Allows you to exit from the program

## **View**

Toolbar Display or hide the toolbar  
Status bar Display or hide the status bar  
Glove Settings Allows you to connect or disconnect the glove  
Landscape Mode Switch the background landscape on or off  
Object  
- 1. None No object loaded  
- 2. Hand Animated hand loaded  
- 3. Dude<sup>3</sup> Animated character loaded

*Please take note that the landscape and an object may be loaded simultaneously*

## **3D Driver**

Bilinear Filtering Switch bilinear filtering on or off  
Dual Texture Switch dual texturing on or off  
Fogging Switch fogging on or off  
Fill  
- Point Shown the landscape/object vertices only  
- Wireframe Show the landscape/object in wireframe mode  
- Solid Show the landscape in solid fill mode  
Open GL Render the landscape/object with Open GL  
DirectX (ver 7.0) Render the landscape/object with DirectX 7.0  
*(DirectX 7.0 needs to be installed)*  
Setup... This option allows the user to change DirectX or OpenGL settings according to which of the two is currently selected.

## **Help**

About About 5DT Data Glove Demo

The following toolbar options are available:



Displays the About Box



Opens the Connect Dialog Box



Allows the user to rotate the viewpoint around the loaded object by left-clicking and moving the mouse simultaneously



Allows the user to move the viewpoint around the loaded object by left-clicking and moving the mouse simultaneously



Allows the user to zoom in and out with respect to the loaded object by left-clicking and moving the mouse simultaneously

---

<sup>3</sup> Dude is a relic from a bygone era. He will be fashionable every 20-30 years.

## Using the Demo

When a particular object or character is chosen, a dialog box will appear with specific instructions on how to animate that specific object and/or move around in the landscape.

### 4.3 Glove Programming Examples

`...\Programming Examples\VC Sample`

Sample Source Code (Microsoft Visual C++ 6.0) is supplied to illustrate how the Glove Driver may be used.

*A Linux programming example is also available and can be found under the  
\`LINUX`\ directory of your CD/Disk*

## 5 Troubleshooting and Support

---

The following information is provided to help you diagnose any problems that may be encountered with your 5DT Data Glove 16 or 16-W. Please carefully consult this section before contacting your supplier.

### 5.1 General Troubleshooting

If the glove is not working at all, please work through the **physical connection problems** section of the troubleshooting procedure to identify the source of the problem.

#### Physical Connection Problems

1. Check that the glove is getting power. A red light (LED) will be shining if the power is adequate. Check that the power supply is 9V DC, center positive.
2. Check that the glove is connected to the right port. Try to connect the glove to another port (perhaps using a 9 to 25 pin adapter) and try again.
3. Check all the connections. Chapter 2 has information on how to correctly connect the interface cable to the glove, the power supply and the computer. Refer to Fig. 4 or Fig. 5.
4. Check the serial cable. Consult the interface cable specifications in Appendix B. You will find that a multimeter is helpful here.
5. Check that the port is working. Try connecting another device, for example an external modem.

If the 5DT Data Glove is recognized by your computer but not connecting properly, proceed directly to the **software connection problems** section.

#### Software Connection Problems

Run **Glove Manager** and select **Connect** from the **Glove** menu. If the **Ports** tab is selected, you will be able to see which COM ports are available. Remember that not all available ports will be physically connected to a socket on the back of your computer. Some ports may be unavailable because other programs have already taken control of the ports. Examples are mouse drivers and fax or communication programs.

Select an available port and click the **Connect** button to search for a glove on that port. If a glove is not detected, one of the following error messages may be returned by Glove Manager:

**Could not open COMx**

Glove Manager attempted to open the specified COM port, but was not successful. This is either because the port does not exist or because another program (or device driver) has already opened the port.

**No response from Glove**

The port could be opened and data could be sent, but no data (or invalid data) was returned by the glove.

**No response from Glove: could be a modem**

The port could be opened and data could be sent. The data that was returned indicates that a modem may be connected to the specified port.

#### Communication failed

A glove could be detected, but was not responding correctly. This could indicate that the glove electronics are not functioning correctly. Try switching the glove off and try again

#### Could not read Glove information

#### Corrupted info header received

These messages indicate that a glove was found and it responded correctly, however the glove information could not be read (or was invalid). Switch the glove off and back on again, and then try again.

#### No data stream from Glove

A working glove was found, however it did not return a stream of data when requested to do so.

## 5.2 Frequently Asked Questions (FAQ)

- Q** My glove is connected to COM2, but Glove Manager reports that this port is not available.
- A** There may be a mouse driver, or other program that has already opened that port. Once a program or driver has opened a COM port, no other program will be able to access that port until the first one closes.
- Q** Is it possible to use the glove in a LINUX environment?
- A** Yes. The 5DT Data Glove Driver is a cross-platform driver and includes a LINUX driver.
- Q** Is it possible to use the glove in a UNIX environment?
- A** Yes. UNIX drivers are available. Please contact 5DT for more information.
- Q** How do I clean my glove?
- A** With a great deal of finesse. The lycra part of the glove may be soaked in cold water with mild detergent, **but only after all the electronics have been disconnected**. This includes the interface box and the optical electronics board. **DO NOT** soak the electronics. **DO NOT** scrub, dry-clean or spin-dry the lycra part. To dry the lycra once it has been soaked, carefully put it down on a towel and allow it to dry. When the lycra has dried, carefully clean the front-ends of the optical fibers with a dry cotton bud before assembling the glove again.
- Q** Can I use four (or more) gloves on one machine at the same time?
- A** 5DT just loves people like you! The number of gloves is limited by the number of serial ports on your computer. Multiport cards are available that have up to eight serial ports on them. It is also possible to obtain a USB (Universal Serial Bus) device that has many RS 232 serial ports on it. The software supplied with the glove will work with any Win32 port setting from COM1 to COM8.
- Q** When I flex a finger, the output does not change but stays at the maximum value. What is the problem?
- A** In this case it is possible that one of the fibers has failed. The glove will then have to be returned to the supplier (or 5DT) for repairs.

### **5.3 Support**

If you experience problems or have complaints, suggestions or other comments, please feel free to contact your supplier (preferably via e-mail). Please have the following information available, or include it in your written correspondence:

1. Computer processor make and model (e.g. Pentium II 350)
2. Operating system and version
3. The symptoms of the problem and what was happening when the problem occurred
4. Can you reproduce the problem easily? Describe how
5. Actions taken by yourself to solve the problem

You may also contact 5DT directly. Please refer to our contact details in section 7.

**Your feedback helps make this a truly great product.**



## 6 Warranty Information

---

### 6.1 Thirty (30) Day Customer Satisfaction Guarantee

If, for any reason, you are disappointed with this product, please contact your supplier. If you are not completely satisfied, you will be granted a full refund once the product is returned undamaged. Unfortunately, shipping costs cannot be refunded.

### 6.2 One (1) Year Product Warranty

5DT warrants to the original purchaser of the 5DT Data Glove 16 or 5DT Data Glove 16-W that it will be free of defects in materials and workmanship for a period of one year from the original date of purchase. During the warranty period, 5DT will repair or replace (with a reconditioned unit) components that are defective.

### Exclusions

The above warranty is provided for private use only and shall not apply to any commercial use of the product, including (but not limited to):

- Location based entertainment (LBE) applications
- Trade show demonstrations

This warranty shall not apply to defects resulting from the following:

- Misuse
- Improper or inadequate maintenance
- Unauthorized modification

### 6.3 Warranty Claim Instructions

Please use the following procedure if you require warranty service:

1. Contact your supplier to determine if you need to return the product. If your supplier is unable to determine this, contact 5DT directly. Do not return a product to your supplier or to 5DT **without first contacting them**.
2. Your supplier (or 5DT) will issue you with a Return Material Authorization (RMA) number. Do not return a product to your supplier or to 5DT without an RMA number.
3. Pack the items to be returned securely using the original packaging material (if possible). Please do not ship the glove in an envelope, use a cardboard box instead.
4. Please enclose the following:
  - Your original sales bill
  - Name, address, contact telephone/fax numbers
  - E-mail address
  - Reason for returning the item
  - RMA number
5. Mark your shipping container with the RMA number to expedite handling at your supplier or at 5DT.
6. Ship prepaid to your supplier or to 5DT.
7. When completing customs or courier documentation, please clearly indicate: "Unit returned for repairs under warranty".

5DT is not responsible for any damage that may occur during shipping. Shipping charges to 5DT are your responsibility. COD shipments will not be accepted by 5DT.

## 7 About 5DT

---

5DT <Fifth Dimension Technologies> is a high technology company specializing in Virtual Reality (VR). 5DT develops, produces and distributes affordable virtual reality hardware, software and systems. 5DT can also develop custom systems for all your VR needs.

5DT's focus is Virtual Reality Training Systems and Virtual Reality Peripherals.

The Virtual Reality hardware currently produced by 5DT includes:

- 5DT Data Glove 5 (Right and left)
- 5DT Data Glove 5-W (Wireless, right and left)
- 5DT Data Glove 16 (Right and left)
- 5DT Data Glove 16-W (Wireless, right and left)
- 5DT Data Glove MRI
- 5DT Head Mounted Display (HMD)
- 5DT 3-DOF Orientation Tracker
- 5DT Space Controller

Some of the Virtual Reality software and technology demonstrators that have been developed by 5DT include:

- Continuous Miner Training Simulator (Mechanized Coal Mining)
- High Voltage Yard Training Simulator
- High Voltage Line Inspection Training Simulator
- Air Traffic Control (ATC) Training Simulator
- Exposure Treatment Phobia Simulator
- Gastroscope Training Simulator
- Crane Operator Training Simulator
- VR FLO VIZ (Virtual Wind Tunnel)
- VR FLIGHT VIZ (Tactical Training Simulator for air combat)
- Integrated Virtual Battlefield
- Rocket Launcher Training Simulator
- Fiber Guided Video Missile Training System
- Landscape Generation and Real-time Rendering Technology
- Unmanned Aerial Vehicle (UAV) Training Simulator

5DT also produces multimedia software. Some of our existing products include:

- High Voltage Line Inspection Training System
- High Voltage Line Repair Training System
- High Voltage Line Visualization System
- Gastroenterology Training System

Most of our systems may be customized to suit specific end-user requirements.

Some of the systems currently under development at 5DT:

- Virtual Binoculars
- Bronchoscope Training Simulator
- Attack Helicopter Tactical Training Simulator

Advance information on these systems may be supplied on request.

## **5DT – USA and Americas**

### Physical & Postal Address:

15375 Barranca Pkwy, G-103  
Irvine, CA 92618, USA

Tel: +1 949 450-9044

Fax: +1 949 450-9045

e-mail: [info.us@5dt.com](mailto:info.us@5dt.com)

Web: [www.5dt.com](http://www.5dt.com)

## **5DT – Rest of World**

### Physical Address (for courier deliveries; not for mail)

25 De Havilland Crescent  
Persequor Technopark  
Pretoria  
South Africa

### Postal Address (mail only)

P.O. Box 5  
Persequor Technopark  
0020 PRETORIA  
South Africa

Tel: +27 12 349 2690

Fax: +27 12 349 1404

Email: [info@5dt.com](mailto:info@5dt.com)

Web: [www.5DT.com](http://www.5DT.com)

## Appendix A - Hardware Specifications

---

- MATERIAL:** Black stretch lycra
- FLEXURE RESOLUTION:** 12-bit A/D for each sensor.  
Minimum dynamic range is 8-bits.
- BEND SENSORS:** Proprietary fiber optic based flexor technology. 2 sensors per finger, 1 sensor between each finger.  
*(Thumb position sensor and wrist flexure sensor not yet implemented)*
- COMPUTER INTERFACE:** Standard glove:  
RS 232 (3-wire)  
GND, TX  
19200 bps (transmit only)  
8 data bits, 1 stop bit, no parity
- Wireless glove:  
RS 232 (2-wire)  
GND, TX  
9600 bps (transmit only)  
8 data bits, 1 stop bit, no parity
- POWER SUPPLY:** Maximum 150 mA @ 9 V DC  
Center positive DC power connector
- SAMPLING RATE:** The full hand (14 sensors) may be sampled at up to 100 samples per second.
- TRANSMISSION FREQUENCY:** Right-handed wireless glove:  
418 MHz
- Left-handed wireless glove:  
433.92 MHz
- TYPICAL WIRELESS RANGE:** Up to 30m

Due to a program of continuous improvement, these specifications are subject to change without prior notice.

## Appendix B - Interface Cable Technical Details

---

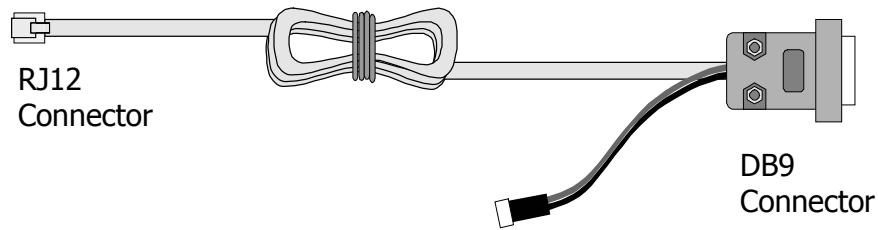


Figure 11 – The interface cable

RJ12 connector (glove side)



Figure 12 – RJ12 Connector

DB9 Connector (Computer Side)

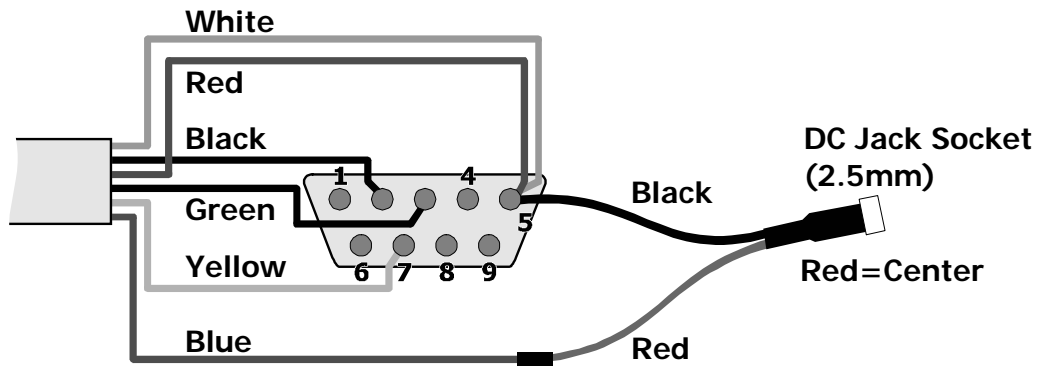


Figure 13 – DB9 Female Connector

**DB9 Pins Used:**

- 2 – RD
- 3 – TD\*
- 5 – GND
- 7 – RTS\* (only for mouse emulation mode)

\* Pins 3 and 7 are only used in the 5DT Data Glove 5

## Appendix C - Serial Port Interface Protocol

---

The data glove is connected to the RS 232 port of the computer. The 5DT Data Glove 16 is a transmit-only device, as it does not respond to any commands sent to it.

### C1. Communications

The 5DT Data Glove 16 has the following serial settings:

- 19200 bits per second
- 8 data bits, 1 stop bit, no parity
- Hardware handshaking: none

The serial link only utilizes the TX and GND lines.

The glove emits two different packets of information. The first packet is transmitted continuously, while the second packet is transmitted occasionally.

An example data stream from the glove could look as follows:

`gdata16 gdata16 gdata16 ginfo16 gdata16 gdata16`

These packets will now be discussed in more detail.

`gdata16`

The measured flexures of each of the sensors are the returned in this packet. The packet consists of 36 bytes and has the following structure:

`header s1high s1low ... s16high s16low checksum trailer`

The `header` is always two bytes long:

`0x3C` (an ASCII "<", decimal value 60)

`0x44` (an ASCII "D", decimal value 68)

After the header, two bytes are sent for each sensor. The high order byte is sent first, then the low order byte. The value of the sensor is therefore  $(s\_high) \times 256 + s\_low$ .

The `checksum` is the least significant byte of the addition of all the sensor values in the packet.

At the end of the packet, a trailing byte with the value of `0x3E` (an ASCII ">", decimal value 62) is sent.

## `ginfo16`

This packet is sent occasionally. It contains information about the glove.

`header` `d1...` `d4` `trailer`

The `header` is always two bytes long:

`0x3C` (an ASCII "<", decimal value 60)

`0x46` (an ASCII "I", decimal value 71)

The four bytes of data will always be defined as follows:

(Byte 1) Version major, for example, the 1 in 1.04

(Byte 2) Version minor, for example, the 04 in 1.04

(Byte 3) Capability word low

bit 0 is set if the glove is a right hand glove

bit 6 is set if the glove is a left hand glove

(Byte 4) Capability word high

bit 0 is set if the glove is a wireless glove

At the end of the packet, a trailing byte with the value of `0x3E` (an ASCII ">", decimal value 62) is sent.

# Appendix D – 5DT Data Glove Driver

---

## 1. Introduction

The 5DT Data Glove Driver provides access to the 5DT range of data gloves at an intermediate level.

The Windows 95/98/NT version is provided in the form of a C/C++ header (.h) file, a Microsoft Visual C++ library (.lib) file and a dynamic link library (.dll) file.

The Linux version is provided in form of a C/C++ header (.h) file, and a dynamic library (.so) file.

The driver functionality includes:

- Multiple instances
- Easy initialization and shutdown
- Basic (raw) sensor values
- Scaled (auto-calibrated) sensor values
- Calibration functions
- Basic gesture recognition
- Cross-Platform Application Programming Interface

## 2. Using the glove driver

The 5DT Data Glove Driver is easy to implement. Use the following guidelines:

### 2.1 Windows 95/98/NT

- i. Make sure that the header file `fglove.h`, the library file `fglove.lib` and the dynamic link library file `fglove.dll` reside in the current (application) directory, or somewhere that they can be found. The file `fglove.dll` may be copied into your Windows system directory.
- ii. Include the header file `fglove.h` in the application where necessary.
- iii. Add the library file `fglove.lib` to the link process.

There is also a debug version of the driver (`fgloved.lib`, `fgloved.dll`) which outputs debug messages to the debugger.

### 2.2 Linux

The driver requires the Linux Posix threads library `libpthread.so`, which is usually installed with Linux. The driver is linked to `libc6`.



### 2.2.1 Installing the library

The driver is a dynamic library file (`libfglove.so`) which must be installed somewhere where applications can find it. If you have root access to the system, the easiest method is to copy the file into the `/usr/lib` directory, e.g.:

```
# cp libfglove.so /usr/lib
```

If you do not have root access to the system, you should place the library somewhere in your home directory, and then set the `LD_LIBRARY_PATH` environment variable to include the full path of the directory in which you have placed the library. This will indicate to the dynamic library loader where to find the file. For example:

```
$ mkdir /home/yourhomedir/libs
$ cp libfglove.so /home/yourhomedir/libs
$ export LD_LIBRARY_PATH="/home/yourhomedir/libs"
```

In the above example, replace "yourhomedir" with your own home directory name.

### 2.2.2 Setting up access to the serial port

The 5DT Data Glove accesses the serial port using the standard Unix/Linux device files in the `/dev` directory. Applications that use the glove should typically give the user the option of specifying which device to check, such as `/dev/ttyS1`. It is recommended that you create a symbolic link `/dev/fglove` to your device file, e.g:

```
# cd /dev
# ln -s ttyS1 fglove
```

This may make the setup of applications easier should you need to change the port that the glove is connected to.

It is also required that the user of the glove has read/write access to the serial port device file. By default, normally only the root user has these rights. If non-root users on the system will be using the glove, the root user must grant access rights to the device file. For example, the following command will give everyone on the system full access to the serial port `/dev/ttyS1`:

```
# chmod 777 /dev/ttyS1
```

### 2.2.3 The header file

The header file (`fglove.h`) should be copied somewhere where your compiler can find it. If you have root access, the easiest is to copy it to `/usr/include`. If not, then just place the header file in the same directory as your application.

## 2.3 Sample source code

Sample source code that makes use of the glove driver is distributed with the driver.

### 3. Supported gloves

The glove driver supports all 5DT Data Gloves. The current version implements 18 possible sensors, and includes the roll and pitch sensors of the original 5DT Data Glove 5. The driver attempts to map values to all sensor outputs. If it is unable to do so the sensor value defaults to a sensible value. This value can be adjusted by forcing a specific value. To the application programmer the driver therefore appears the same regardless of the type of glove that is connected.

#### 3.1 Sensor mappings for the 5DT Data Glove 5

The sensors on the 5DT Data Glove 5 are positioned as in Fig. 14.

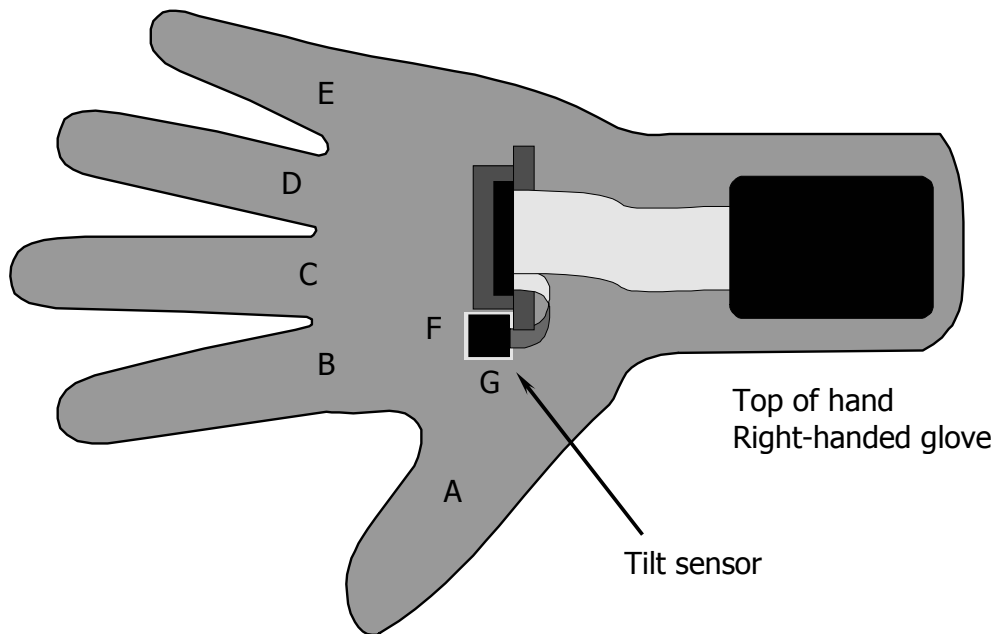


Figure 14 - Sensor positions for the 5DT Data Glove 5

<i>Sensor</i>	<i>Driver Index</i>	<i>Sensor</i>	<i>Description</i>
A	0,1*		Thumb flexure
B	3,4*		Index finger flexure
C	6,7*		Middle finger flexure
D	9,10*		Ring finger flexure
E	12,13*		Little finger flexure
F	16		Pitch angle of tilt sensor
G	17		Roll angle of tilt sensor

Table 2 - Sensor mappings for 5DT Data Glove 5

\* Both these driver sensor indices will return the same value when the 5DT Data Glove 5 is used.

### 3.2 Sensor mappings for the 5DT Data Glove 16

The sensors on the 5DT Data Glove 16 are positioned as in Fig. 15

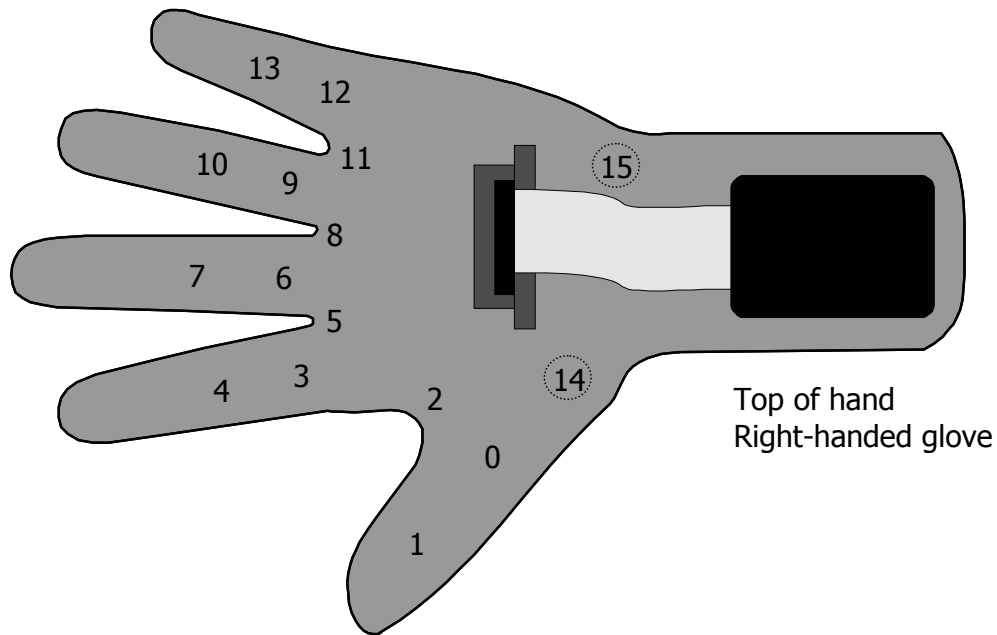


Figure 15 - Sensor mappings for the 5DT Data Glove 16

<i>Sensor</i>	<i>Driver Index</i>	<i>Sensor</i>	<i>Description</i>
0	0		Thumb flexure ( lower joint )
1	1		Thumb flexure ( second joint )
2	2		Thumb-index finger abduction
3	3		Index finger flexure ( at knuckle )
4	4		Index finger flexure ( second joint )
5	5		Index-middle finger abduction
6	6		Middle finger flexure ( at knuckle )
7	7		Middle finger flexure ( second joint )
8	8		Middle-ring finger abduction
9	9		Ring finger flexure ( at knuckle )
10	10		Ring finger flexure ( second joint )
11	11		Ring-little finger abduction
12	12		Little finger flexure ( at knuckle )
13	13		Little finger flexure ( second joint )
14	14		Thumb translation [not yet implemented]
15	15		Wrist flexure [not yet implemented]

Table 3 - Sensor mappings for 5DT Data Glove 16

## 4. Function reference

The following functions are provided by the driver:

---

```
fdGlove *fdOpen(char *pPort)
```

Initializes the glove device on the specified port.

### Return value

Returns a pointer to the glove device (`fdGlove *`). NULL is returned if an error occurred.

### Parameter

`pPort`

Pointer to a zero terminated ASCII string containing the name of the communication port. Valid values on Windows range from "COM1" to "COM8". Unix/Linux port names will differ.

### Remarks

Do not attempt to alter the contents of the returned pointer directly, use the functions provided instead.

---

```
int fdClose(fdGlove *pFG)
```

Frees the glove device and communications port.

### Return value

Returns nonzero if successful, zero if an error occurred.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

### Remarks

It is important to call this function when you are finished using the glove.

---

```
int fdGetGloveHand(fdGlove *pFG)
```

Obtains the handedness (left or right handed) of the glove.

### Return value

Returns either `FD_HAND_LEFT` or `FD_HAND_RIGHT`, as defined by the enumerated type `EfdGloveHand`.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

### Remarks

None.

---

```
int fdGetGloveType(fdGlove *pFG)
```

Obtains the type of the currently connected glove.

**Return value**

Returns one of `FD_GLOVENONE`, `FD_GLOVE7`, `FD_GLOVE7W`, `FD_GLOVE16` or `FD_GLOVE16W`, as defined by the enumerated type `EfdGloveTypes`.

**Parameters**

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

**Remarks**

`FD_GLOVE7` and `FD_GLOVE7W` refer to the original 5+2 (tilt angles) sensor glove (5DT Data Glove 5). The *w* suffix indicates a wireless model. `FD_GLOVE16` and `FD_GLOVE16W` refer to the 16 sensor glove. In order to accommodate both glove types the `fdGetNumSensors()` function currently returns 18 sensors. The additional two sensors are defined as the original tilt angles that are not present in the 16-sensor glove. See the description of `fdGetNumSensors()` for more details.

---

```
int fdGetNumSensors(fdGlove *pFG)
```

Obtains the number of available sensors values the driver can make available.

**Return value**

Returns the number of sensors. Currently it is fixed at 18, but future driver releases may differ.

**Parameters**

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

**Remarks**

Although the 5-sensor glove can measure only average flexure, the driver will attempt to fill in missing values. The number of sensors returned can therefore be of a higher dimension. The enumerated type `EfdSensors` defines the finger mapping for each sensor.

---

```
void fdGetSensorRawAll(fdGlove *pFG, unsigned short *pData)
```

Obtains the most recent raw sensor values from the currently connected glove.

**Return value**

None.

**Parameters**

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`pData`

Pointer to an array of 16-bit integers that will contain the raw sensor values. The size of the array must always match the value returned by `fdGetNumSensors()`.

### Remarks

Currently the raw sensor samples are all 12 bit unsigned values. The range is therefore from 0 to 4095. Note that this is not the dynamic range of the sensors. There can be severe offset values associated with each sensor. The enumerated type `EfdSensors` defines the finger mapping for each sensor.

---

```
unsigned short fdGetSensorRaw(fdGlove *pFG, int nSensor)
```

Obtains the most recent raw sensor value for a specific sensor from the currently connected glove.

### Return value

Returns a 16-bit integer. See `fdGetSensorRawAll()` for details.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`nSensor`

Index of the sensor that is being set. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

### Remarks

The enumerated type `EfdSensors` defines the finger mapping for each sensor.

---

```
void fdSetSensorRawAll(fdGlove *pFG, unsigned short *pData)
```

Forces the raw value for all the sensors.

### Return value

None.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`pData`

Pointer to an array of 16-bit integers that will contain the raw sensor values. The size of the array must always match the value returned by `fdGetNumSensors()`.

### Remarks

Currently the raw sensor samples are all 12 bit unsigned values. The range is therefore from 0 to 4095. The enumerated type `EfdSensors` defines the finger mapping for each

sensor. Forcing a sensor value will result in a raw and scaled output other than the default zero. Values that can be mapped will be overwritten, rendering the forced value void.

---

```
void fdSetSensorRaw(fdGlove *pFG, int nSensor, unsigned short nRaw)
```

Forces the raw value for a specific sensor.

### Return value

None.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`nSensor`

Index of the sensor that is being set. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

`nRaw`

16 bit raw value of the sensor. If the sensor is unmapped, the scaling calculations will proceed as normal.

### Remarks

The enumerated type `EfdSensors` defines the finger mapping for each sensor. This function is only useful for sensors that cannot be mapped by a specific hardware device. Forcing a sensor value will result in a raw and scaled output other than the default zero. Values that can be mapped will be overwritten, rendering the forced value void.

---

```
void fdGetSensorScaledAll(fdGlove *pFG, float *pData)
```

Obtains the most recent scaled (auto-calibrated) sensor values from the currently connected glove.

### Return value

None.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`pData`

Pointer to an array of floating point numbers that will contain the scaled sensor values. The size of the array must always match the value returned by `fdGetNumSensors()`.

### Remarks

The sensor range is a value from zero to the value defined by the `fdSetSensorMax()` and `fdSetSensorMaxAll()` functions. The glove driver defaults to a range of [0...1].

The automatic calibration process is described in section 6. The enumerated type `EfdSensors` defines the finger mapping for each sensor.

---

```
float fdGetSensorScaled(fdGlove *pFG, int nSensor)
```

Obtains the most recent scaled (auto-calibrated) value for a specific sensor from the currently connected glove.

**Return value**

Returns a floating point sensor value.

**Parameters**

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`nSensor`

Index of the sensor that is being set. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

**Remarks**

The sensor range is a value from zero to the value defined by the `fdSetSensorMax()` and `fdSetSensorMaxAll()` functions. The glove driver defaults to a range of [0...1]. The automatic calibration process is described in section 6. The enumerated type `EfdSensors` defines the finger mapping for each sensor.

---

```
int fdGetNumGestures(fdGlove *pFG)
```

Obtains the number of available gestures that can be recognized by the glove driver.

**Return value**

Returns the number of available gestures. Currently 16 different gestures are supported. Refer to section 5 for details.

**Parameters**

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

**Remarks**

None.

---

```
int fdGetGesture(fdGlove *pFG)
```

Obtains the current gesture being performed.

**Return value**

Returns the current gesture being performed. Refer to section 5 for details.



## Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

## Remarks

None.

---

```
void fdGetCalibrationAll(fdGlove *pFG, unsigned short *pUpper,
unsigned short *pLower)
```

Obtains the current auto-calibration settings of the driver.

## Return value

None.

## Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`pUpper` and `pLower`

Arrays of 16 bit unsigned integers that will contain the maximum and minimum raw sensor values. The size of each array must always match the value returned by `fdGetNumSensors()`. Refer to section 6 for details.

## Remarks

None.

---

```
void fdGetCalibration(fdGlove *pFG, int nSensor, unsigned short
*pUpper, unsigned short *pLower)
```

Obtains the current auto-calibration settings of the driver for a specific sensor.

## Return value

None.

## Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`nSensor`

Index of the sensor that is being set. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

`pUpper` and `pLower`

Pointers to 16 bit unsigned integers containing the maximum and minimum raw sensor values. Refer to section 6 for details.

## Remarks

None.

---

```
void fdSetCalibrationAll(fdGlove *pFG, unsigned short *pUpper,  
unsigned short *pLower)
```

Resets the current auto-calibration settings of the driver to user defined values.

### Return value

None.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`pUpper` and `pLower`

Arrays of 16 bit unsigned integers containing the maximum and minimum raw sensor values. The size of each array must always match the value returned by `fdGetNumSensors()`. Refer to section 6 for details.

### Remarks

For unmapped sensors it would be sensible to set the upper and lower calibration settings above and below the raw value forced with `fdSetSensorRaw()` and `fdSetSensorRawAll()`.

---

```
void fdSetCalibration(fdGlove *pFG, int nSensor, unsigned short  
nUpper, unsigned short nLower)
```

Resets the current auto-calibration settings of the driver for a specific sensor to user defined values.

### Return value

None.

### Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`nSensor`

Index of the sensor that is being set. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

`nUpper` and `nLower`

16 bit unsigned integers containing the maximum and minimum raw sensor values. Refer to section 6 for details.

### Remarks

For unmapped sensors it would be sensible to set the upper and lower calibration settings above and below the raw value forced with `fdSetSensorRaw()` and `fdSetSensorRawAll()`.

---

```
void fdResetCalibration(fdGlove *pFG)
```

Resets the internal auto-calibration settings of the driver to appropriate default values.

**Return value**

None.

**Parameters**

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

**Remarks**

This function is similar to the `fdSetCalibrationAll()` function with each of the upper and lower calibration array values set to 0 and 4095 respectively. This function, or any of the other calibration functions, should be called whenever the application starts up or the glove changes users during run-time. For unmapped sensors the upper and lower calibration values are set to 4095 and 0 respectively, which is the *inverse* of the auto-calibration settings.

---

```
void fdGetSensorMaxAll(fdGlove *pFG, float *pMax)
```

Obtains the maximum scaled value for each sensor.

**Return value**

None.

**Parameters**

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`pMax`

Array of floating point values that will contain the maximum scaled sensor values. The size of the array must always match the value returned by `fdGetNumSensors()`.

**Remarks**

The glove driver defaults to a maximum scaled value of 1 for each sensor.

---

```
float fdGetSensorMax(fdGlove *pFG, int nSensor)
```

Obtains the maximum scaled value for a specific sensor.

**Return value**

Returns the maximum scaled values of the sensor.

## Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`nSensor`

Index of the sensor that is being queried. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

## Remarks

The glove driver defaults to a maximum scaled value of 1 for each sensor.

---

```
void fdSetSensorMaxAll(fdGlove *pFG, float *pMax)
```

Sets the maximum scaled value for each sensor.

## Return value

None.

## Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`pMax`

Array of floating point values that contains the maximum scaled sensor values. The size of the array must always match the value returned by `fdGetNumSensors()`.

## Remarks

The glove driver defaults to a maximum scaled value of 1 for each sensor.

---

```
void fdSetSensorMax(fdGlove *pFG, int nSensor, float fMax)
```

Sets the maximum scaled value for a specific sensor.

## Return value

None.

## Parameters

`pFG`

Pointer to a glove device. This is the value returned by `fdOpen()`.

`nSensor`

Index of the sensor that is being set. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

`fMax`

A floating point value that contains the maximum scaled sensor value.

## Remarks

The glove driver defaults to a maximum scaled value of 1 for each sensor.

---

```
void fdGetThresholdAll(fdGlove *pFG, float *pUpper, float *pLower)
```

Obtains the current gesture recognition threshold settings of the driver.

## Return value

None.

## Parameters

*pFG*

Pointer to a glove device. This is the value returned by `fdOpen()`.

*pUpper* and *pLower*

Arrays of floating point numbers that will contain the maximum and minimum threshold values. The size of each array must always match the value returned by `fdGetNumSensors()`. Refer to section 6 for details.

## Remarks

None.

---

```
void fdGetThreshold(fdGlove *pFG, int nSensor, float *pUpper,  
float *pLower)
```

Obtains the current gesture recognition threshold settings of the driver for a specific sensor.

## Return value

None.

## Parameters

*pFG*

Pointer to a glove device. This is the value returned by `fdOpen()`.

*nSensor*

Index of the sensor that is being queried. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

*pUpper* and *pLower*

Pointers to floating point numbers that will contain the maximum and minimum threshold values. Refer to section 6 for details.

## Remarks

None.

---

```
void fdSetThresholdAll(fdGlove *pFG, float *pUpper, float *pLower)
```

Sets the current gesture recognition threshold settings of the driver.

**Return value**

None.

**Parameters**

*pFG*

Pointer to a glove device. This is the value returned by `fdOpen()`.

*pUpper* and *pLower*

Arrays of floating point numbers that contains the maximum and minimum threshold values. The size of each array must always match the value returned by `fdGetNumSensors()`. Refer to section 6 for details.

**Remarks**

None.

---

```
void fdSetThreshold(fdGlove *pFG, int nSensor, float fUpper, float fLower)
```

Sets the current gesture recognition threshold settings of the driver for a specific sensor.

**Return value**

None.

**Parameters**

*pFG*

Pointer to a glove device. This is the value returned by `fdOpen()`.

*nSensor*

Index of the sensor that is being set. The value must lie in the range given by the enumerated type `EfdSensors`, or alternatively from zero to the value returned by `fdGetNumSensors()` minus one.

*fUpper* and *fLower*

Floating point numbers that contain the maximum and minimum threshold values. Refer to section 6 for details.

**Remarks**

None.

---

```
void fdGetGloveInfo(fdGlove *pFG, unsigned char *pData)
```

Obtains the information data block of the currently connected glove.

**Return value**

None.

## Parameters

pFG

Pointer to a glove device. This is the value returned by `fdOpen()`.

pData

Array of 32 bytes that will contain the information data.

## Remarks

The information data is specified in the glove user's manual. The size of the information block is always 32 bytes.

---

```
void fdGetDriverInfo(fdGlove *pFG, unsigned char *pData)
```

Obtains the information data block of the driver.

## Return value

None.

## Parameters

pFG

Pointer to a glove device. This is the value returned by `fdOpen()`.

pData

Array of 32 bytes that will contain the information data.

## Remarks

The information data is a zero terminated string that contains driver information. The size of the information block is always 32 bytes.

## 5. Gesture definitions

The currently defined set of gestures is comprised of binary open/close configurations of the fingers excluding the thumb. There are  $2^4 = 16$  such possible combinations. Gesture number 0 is defined as all the fingers (excluding the thumb) being closed, and gesture number 15 as all the fingers open. The index finger indicates the least significant bit. For example, the index finger point gesture will therefore be number 1, and the little finger point gesture number will be 8. An invalid (unrecognizable) gesture is defined as the value -1.

A scaled sensor value of higher than the upper threshold setting will indicate a closed finger, while a scaled sensor value of lower than the lower threshold setting will indicate an open finger. A value in-between is invalid and will result in an invalid gesture. In the case of multiple finger joint angle measurements (such as the 16 sensor glove), the maximum of the individual joint sensor values is taken to obtain a closed gesture and the minimum to obtain an open gesture. Closed gestures take precedence, in other words, bending only one joint of a finger will count as a closed gesture.

The built-in gesture recognition capabilities of the glove driver is limited in scope and user independence. High level gesture recognition algorithms that rely on training sequences are suggested for advanced applications.

<b>Finger:</b>	<b>Little</b>	<b>Ring</b>	<b>Middle</b>	<b>Index</b>
5DT Data Glove 5 sensor:	E	D	C	B
5DT Data Glove 16 sensor:	12,13	9,10	6,7	3,4
Driver sensor index:	12,13 <sup>#</sup>	9,10 <sup>#</sup>	6,7 <sup>#</sup>	3,4 <sup>#</sup>

<b>Gesture Number</b>	<b>Flexure (0=flexed, 1=unflexed)</b>				<b>Gesture Description</b>	<b>Fig.</b>
0	0	0	0	0	Fist	16.0
1	0	0	0	1	Index finger point	16.1
2	0	0	1	0	Up yours	16.2
3	0	0	1	1	Two finger point	16.3
4	0	1	0	0	Ring finger point	16.4
5	0	1	0	1	Ring index point	16.5
6	0	1	1	0	Ring middle point	16.6
7	0	1	1	1	Three finger point ( or not little point )	16.7
8	1	0	0	0	Little finger point	16.8
9	1	0	0	1	Howzit	16.9
10	1	0	1	0	Little middle point	16.10
11	1	0	1	1	Not ring finger point	16.11
12	1	1	0	0	Little ring point	16.12
13	1	1	0	1	Not up yours	16.13
14	1	1	1	0	Not index finger point	16.14
15	1	1	1	1	Flat hand	16.15

**Table 4 - Gesture definition scheme as implemented for the 5DT Data Glove Driver**

<sup>#</sup> When the 5DT Data Glove 5 is used, both these driver sensor indices will return the same value.

When the 5DT Data Glove 16 is used, the driver sensor indices will return different values. The maximum of the two values will be used to test for a flexed (closed) gesture, and the minimum value of the two will be used to test for an unflexed (open) gesture.



The following gestures are currently defined (right hand shown):



**Figure 16 - Gesture Illustrations**

## 6. Auto-calibration

The driver can provide sensor outputs in an automatic, linearly calibrated fashion. During every update, the raw value read from the sensor is compared to the current minimum and maximum raw values ( $raw_{min}$  and  $raw_{max}$ ) as set by the functions `fdSetCalibrationAll()`, `fdSetCalibration()` or `fdResetCalibration()`. If the current minimum and maximum values are exceeded, they are overwritten. The upper and lower calibration values are therefore continuously pushed "outwards". The normalized output is given by the first order equation

$$out = \frac{raw_{val} - raw_{min}}{raw_{max} - raw_{min}} \cdot Max, \quad (F-1)$$

which is in  $[0..Max]$ . The value of  $Max$  is set by the functions `fdSetSensorMaxAll()` and `fdSetSensorMax()`. Doing a few flexing movements with the hand quickly sets the operating values for  $raw_{min}$  and  $raw_{max}$ , and calibrates the glove.

The auto-calibration process can be ignored by simply regarding only the raw sensor outputs. It would be up to the application developer to provide a suitable calibration process. Note that calibration is *mandatory*, especially with the 16 sensor glove which contains no hardware calibration possibilities (miniature preset potentiometers).

## Appendix E – Software Calibration

---

When the 5DT Data Glove is calibrated in the factory, we try to obtain the largest possible dynamic range. The dynamic range is the difference between the maximum output value (flexed hand) and the minimum output value (flat hand).

$$\text{Dynamic Range} = \text{Value}_{\max} - \text{Value}_{\min} \quad (\text{F-2})$$

The dynamic range may differ when persons with different hand sizes use the glove. We therefore recommend that a software calibration routine be included with all glove applications/programs.

Software calibration normalizes the effect of different dynamic ranges for different hand sizes.

The software calibration routine should store the minimum and maximum values and scale the output for maximum dynamic range. For example, lets work through an example for the thumb.

The minimum value is:	40	=	$\text{Value}_{\min}$
The maximum value is:	206	=	$\text{Value}_{\max}$
The dynamic range is:	$206 - 40$	=	166

To scale the measured values across the full dynamic range (256 values), one therefore has to perform the following operation:

$$\text{Value}_{\text{scaled}} = (\text{Value}_{\text{measured}} - \text{Value}_{\min}) \times (255/\text{Dynamic Range}) \quad (\text{F-3})$$

For 40 this becomes  $(40 - 40) \times (255/166) = 0$

For 206 this becomes  $(206 - 40) \times (255/166) = 255$

This routine could also be implemented dynamically, where the glove calibrates itself while the user uses it. The routine is similar to the normal calibration scheme with the difference that the minimum and maximum values are updated continuously. The initial values for the minimum and maximum could be chosen as 126 and 128 respectively.

*Please note that the 5DT Data Glove Driver includes a dynamic software calibration routine*